FIGURE 1

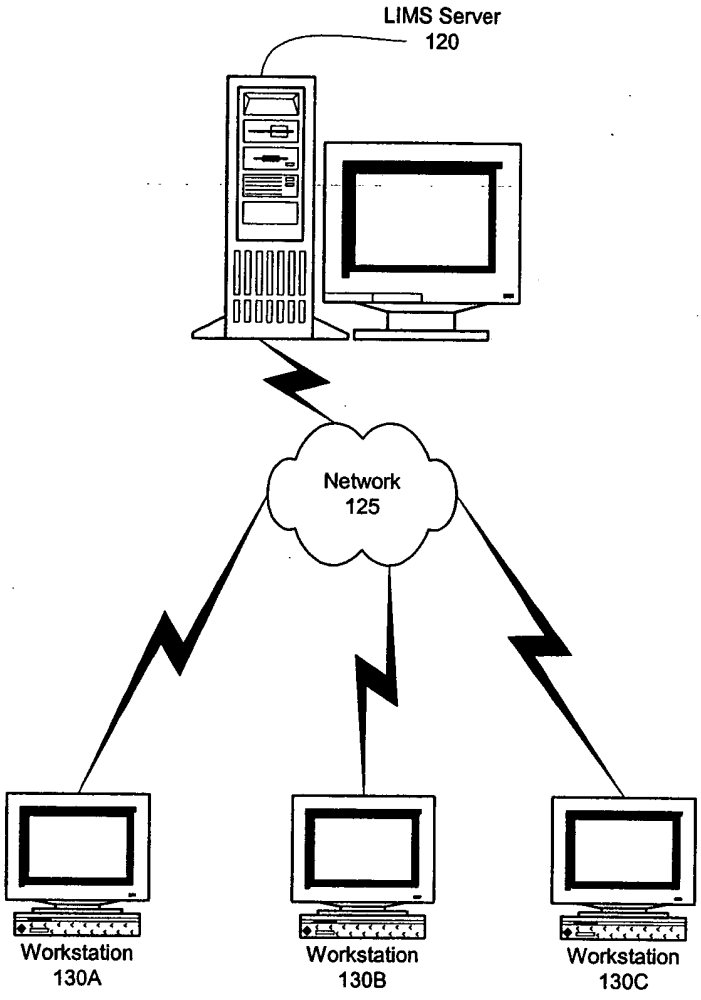

LIMS Server
120
Network
125
Workstation
130A
Workstation
130B
Workstation
130C

FIGURE 2

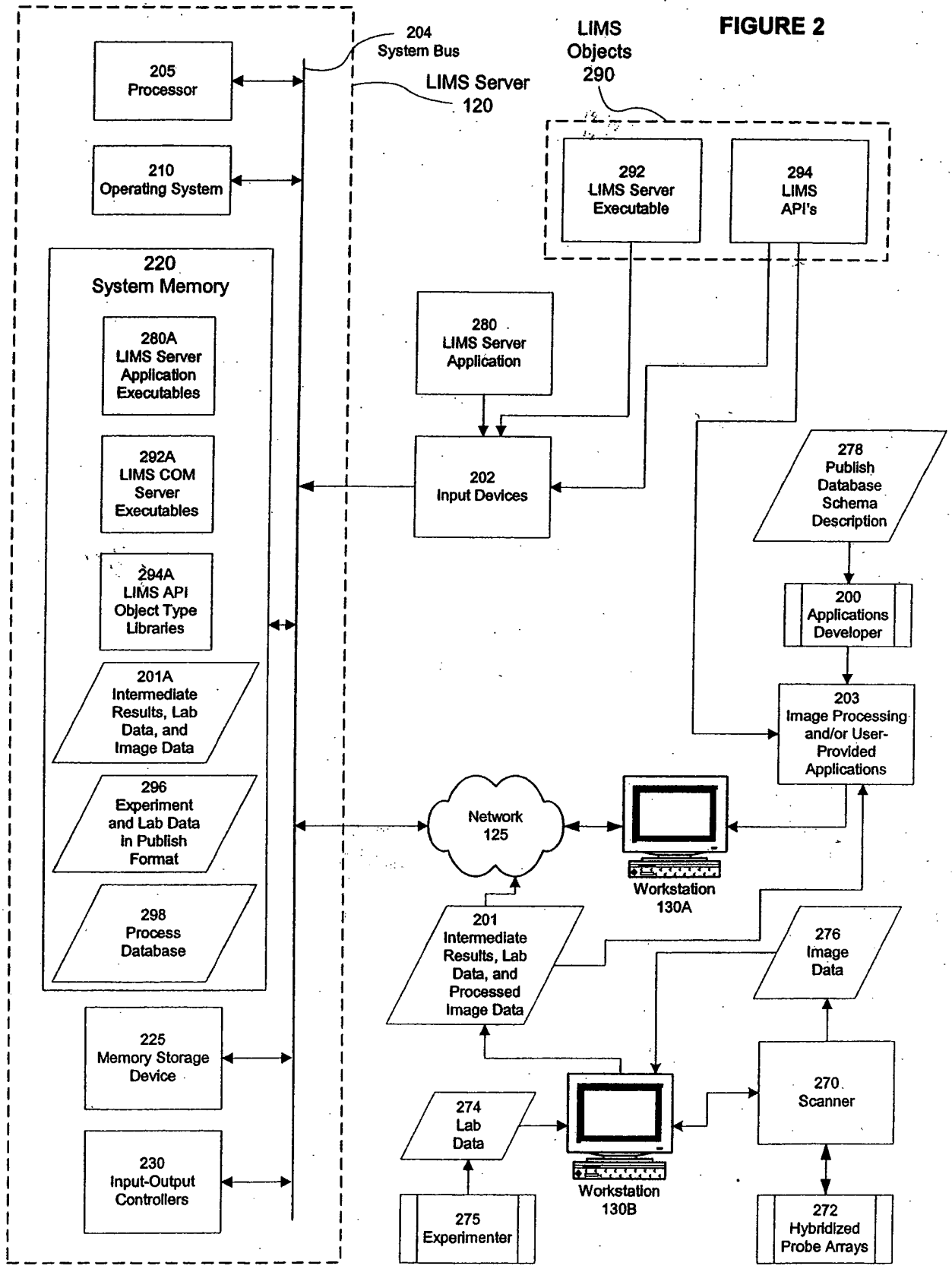

205
Processor
204
System Bus
LIMS Server
120
LIMS
Objects
290
210
Operating System
292
LIMS Server
Executable
294
LIMS
API's
220
System Memory
280A
LIMS Server
Application
Executables
280
LIMS Server
Application
292A
LIMS COM
Server
Executables
202
Input Devices
278
Publish
Database
Schema
Description
294A
LIMS API
Object Type
Libraries
200
Applications
Developer
201A
Intermediate
Results, Lab
Data, and
Image Data
203
Image Processing
and/or User-
Provided
Applications
296
Experiment
and Lab Data
in Publish
Format
Network
125
Workstation
130A
298
Process
Database
201
Intermediate
Results, Lab
Data, and
Processed
Image Data
276
Image
Data
225
Memory Storage
Device
270
Scanner
274
Lab
Data
230
Input-Output
Controllers
Workstation
130B
275
Experimenter
272
Hybridized
Probe Arrays

FIGURE 3

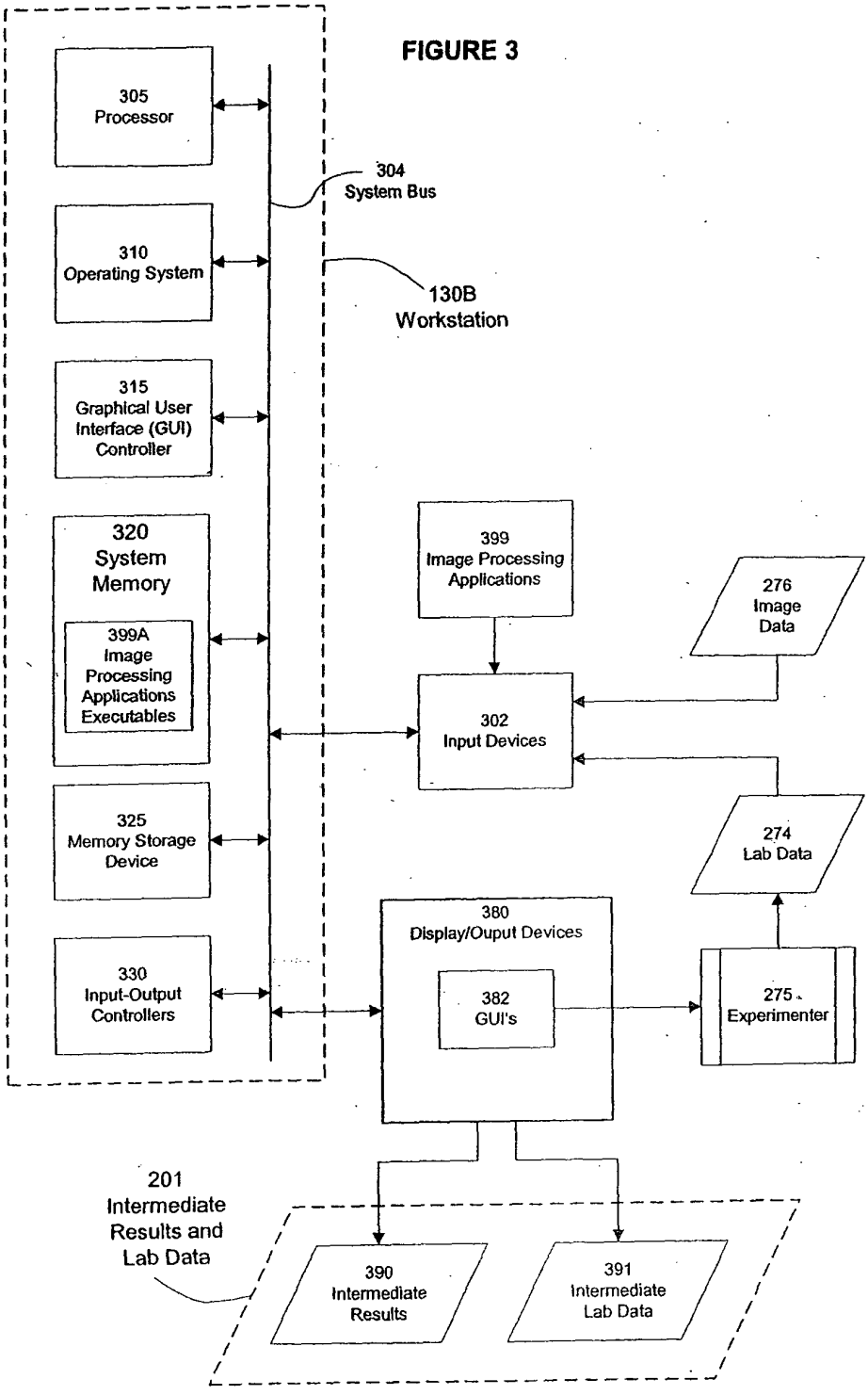
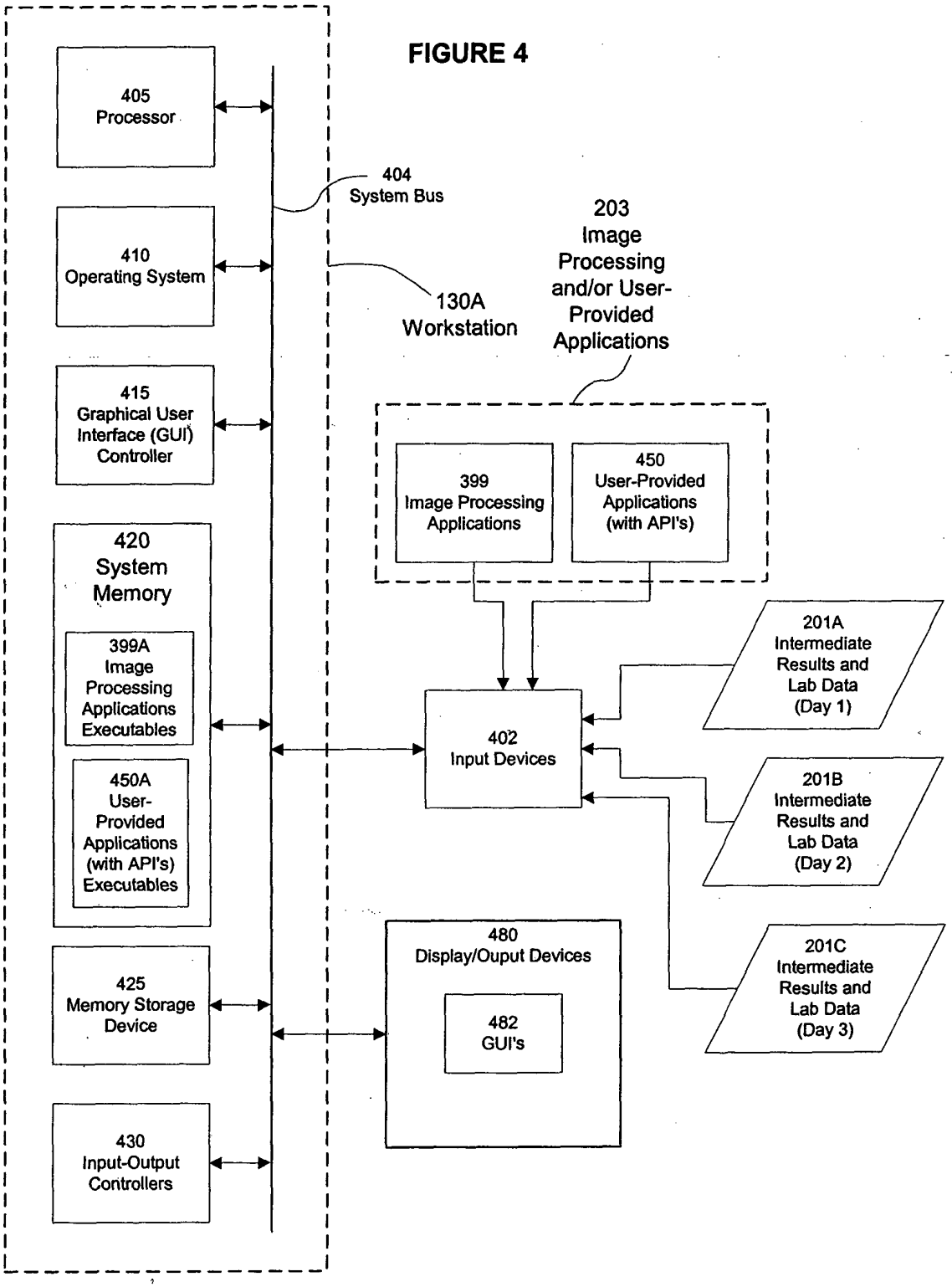

305
Processor
304
System Bus
310
Operating System
130B
Workstation
315
Graphical User
Interface (GUI)
Controller
320
System
Memory
399A
Image
Processing
Applications
Executables
399
Image Processing
Applications
276
Image
Data
302
Input Devices
325
Memory Storage
Device
274
Lab Data
380
Display/Ouput Devices
330
Input-Output
Controllers
382
GUI's
275
Experimenter
201
Intermediate
Results and
Lab Data
390
Intermediate
Results
391
Intermediate
Lab Data

FIGURE 4


405
Processor
404
System Bus
410
Operating System
130A
Workstation
203
Image
Processing
and/or User-
Provided
Applications
415
Graphical User
Interface (GUI)
Controller
399
Image Processing
Applications
450
User-Provided
Applications
(with API's)
420
System
Memory
399A
Image
Processing
Applications
Executables
450A
User-
Provided
Applications
(with API's)
Executables
201A
Intermediate
Results and
Lab Data
(Day 1)
402
Input Devices
201B
Intermediate
Results and
Lab Data
(Day 2)
480
Display/Ouput Devices
482
GUI's
201C
Intermediate
Results and
Lab Data
(Day 3)
425
Memory Storage
Device
430
Input-Output
Controllers

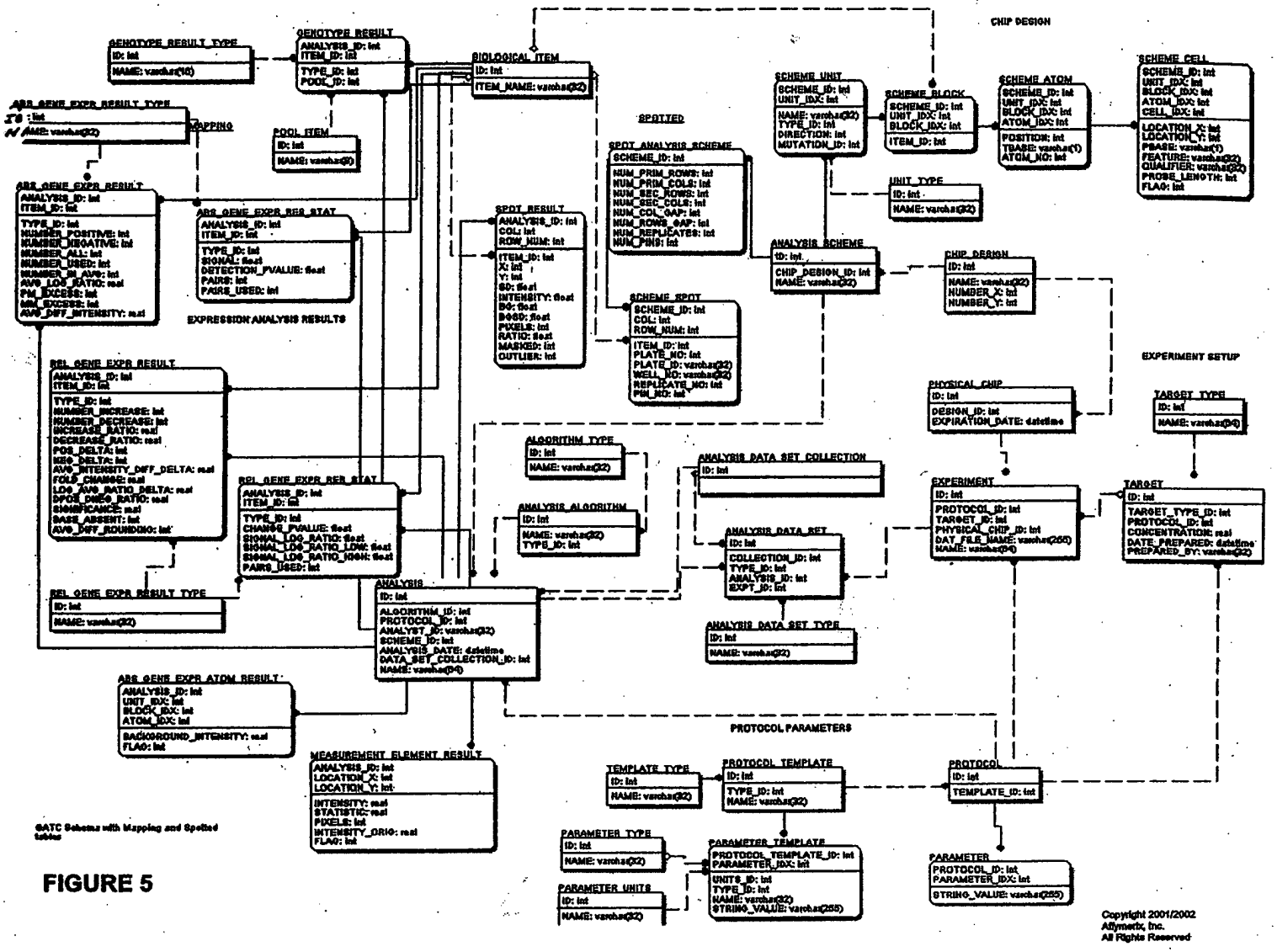
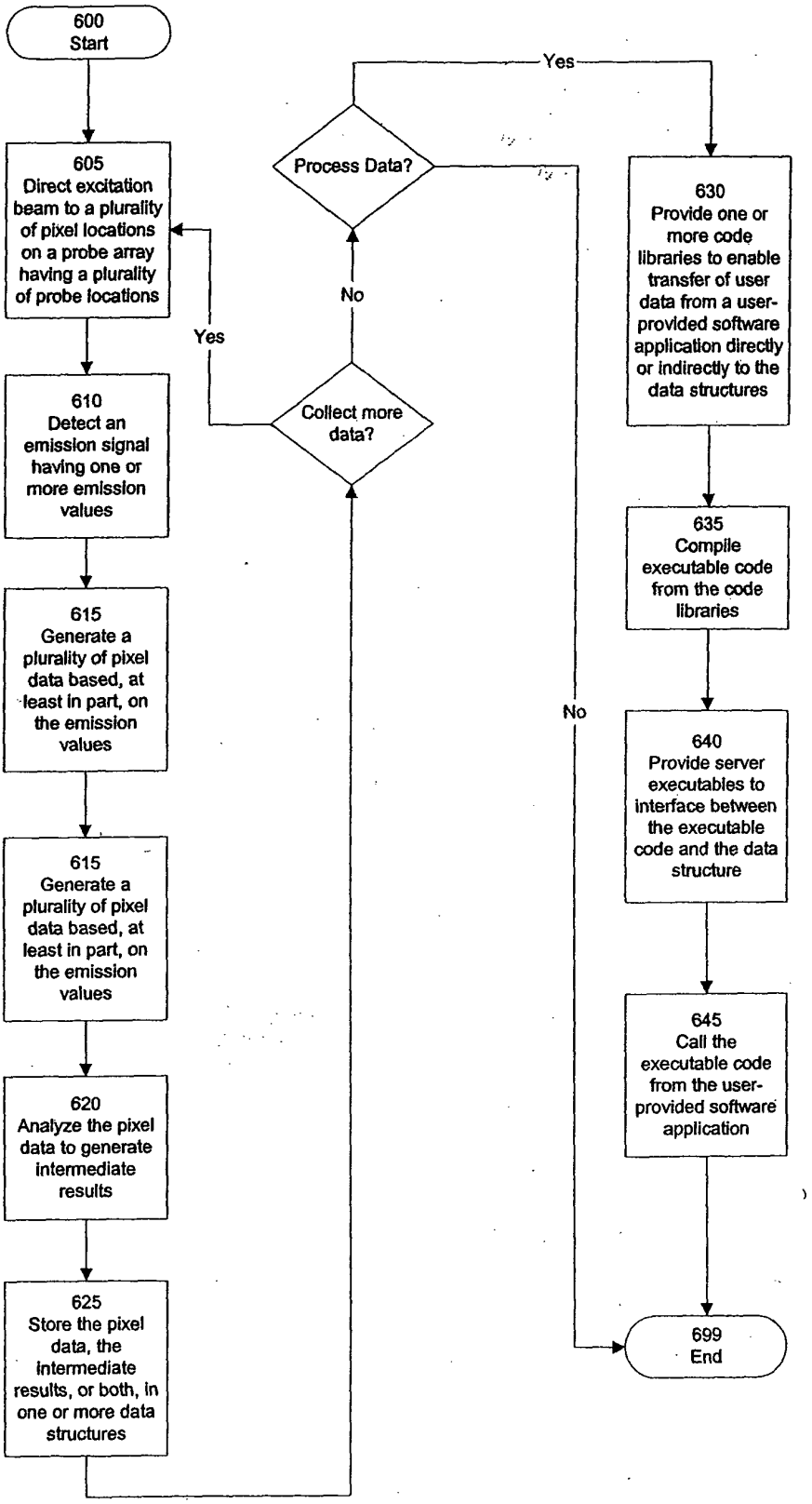

CHIP DESIGN
GENOTYPE_RESULT_TYPE
ID: int
NAME: varchar(16)
GENOTYPE_RESULT
ANALYSIS_ID: int
ITEM_ID: int
TYPE_ID: int
POOL_ID: int
BIOLOGICAL_ITEM
ID: int
ITEM_NAME: varchar(32)
SCHEME_UNIT
SCHEME_ID: int
UNIT_IDX: int
NAME: varchar(32)
TYPE_ID: int
DIRECTION: int
MUTATION_ID: int
SCHEME_BLOCK
SCHEME_ID: int
UNIT_IDX: int
BLOCK_IDX: int
ITEM_ID: int
SCHEME_ATOM
SCHEME_ID: int
UNIT_IDX: int
BLOCK_IDX: int
ATOM_IDX: int
POSITION: int
TRANS: varchar(1)
ATOM_NO: int
SCHEME_CELL
SCHEME_ID: int
UNIT_IDX: int
BLOCK_IDX: int
ATOM_IDX: int
CELL_IDX: int
LOCATION_X: int
LOCATION_Y: int
PBASE: varchar(1)
FEATURE: varchar(32)
QUALIFIER: varchar(32)
PROBE_LENGTH: int
FLAG: int
ABS_GENE_EXPR_RESULT_TYPE
ID: int
NAME: varchar(32)
MAPPING
POOL_ITEM
ID: int
NAME: varchar(8)
SPOTTED
SPOT_ANALYSIS_SCHEME
SCHEME_ID: int
NUM_PRIM_ROWS: int
NUM_PRIM_COLS: int
NUM_SEC_ROWS: int
NUM_SEC_COLS: int
NUM_COL_GAP: int
NUM_ROWS_GAP: int
NUM_REPLICATES: int
NUM_PINS: int
UNIT_TYPE
ID: int
NAME: varchar(32)
ABS_GENE_EXPR_RESULT
ANALYSIS_ID: int
ITEM_ID: int
TYPE_ID: int
NUMBER_POSITIVE: int
NUMBER_NEGATIVE: int
NUMBER_ALL: int
NUMBER_USED: int
NUMBER_IN_AVG: int
AVG_LOG_RATIO: real
PM_EXCESS: int
MM_EXCESS: int
AVG_DIFF_INTENSITY: real
ABS_GENE_EXPR_RES_STAT
ANALYSIS_ID: int
ITEM_ID: int
TYPE_ID: int
SIGNAL: float
DETECTION_PVALUE: float
PAIRS: int
PAIRS_USED: int
EXPRESSION ANALYSIS RESULTS
SPOT_RESULT
ANALYSIS_ID: int
COL: int
ROW_NUM: int
ITEM_ID: int
X: int
Y: int
SD: float
INTENSITY: float
BG: float
BGSD: float
PIXELS: int
RATIO: float
MASKED: int
OUTLIER: int
SCHEME_SPOT
SCHEME_ID: int
COL: int
ROW_NUM: int
ITEM_ID: int
PLATE_NO: int
PLATE_ID: varchar(32)
WELL_NO: varchar(32)
REPLICATE_NO: int
PIN_NO: int
ANALYSIS_SCHEME
ID: int
CHIP_DESIGN_ID: int
NAME: varchar(32)
CHIP_DESIGN
ID: int
NAME: varchar(32)
NUMBER_X: int
NUMBER_Y: int
EXPERIMENT SETUP
REL_GENE_EXPR_RESULT
ANALYSIS_ID: int
ITEM_ID: int
TYPE_ID: int
NUMBER_INCREASE: int
NUMBER_DECREASE: int
INCREASE_RATIO: real
DECREASE_RATIO: real
POS_DELTA: int
NEG_DELTA: int
AVG_INTENSITY_DIFF_DELTA: real
FOLD_CHANGE: real
LOG_AVG_RATIO_DELTA: real
DPOS_DNEG_RATIO: real
SIGNIFICANCE: real
BASE_ABSENT: int
AVG_DIFF_ROUNDING: int
REL_GENE_EXPR_RES_STAT
ANALYSIS_ID: int
ITEM_ID: int
TYPE_ID: int
CHANGE_PVALUE: float
SIGNAL_LOG_RATIO: float
SIGNAL_LOG_RATIO_LOW: float
SIGNAL_LOG_RATIO_HIGH: float
PAIRS_USED: int
ALGORITHM_TYPE
ID: int
NAME: varchar(32)
ANALYSIS_ALGORITHM
ID: int
NAME: varchar(32)
TYPE_ID: int
ANALYSIS_DATA_SET_COLLECTION
ID: int
ANALYSIS_DATA_SET
COLLECTION_ID: int
TYPE_ID: int
ANALYSIS_ID: int
EXPT_ID: int
ANALYSIS_DATA_SET_TYPE
ID: int
NAME: varchar(32)
PHYSICAL_CHIP
ID: int
DESIGN_ID: int
EXPIRATION_DATE: datetime
TARGET_TYPE
ID: int
NAME: varchar(64)
EXPERIMENT
ID: int
PROTOCOL_ID: int
TARGET_ID: int
PHYSICAL_CHIP_ID: int
DAT_FILE_NAME: varchar(255)
NAME: varchar(64)
TARGET
ID: int
TARGET_TYPE_ID: int
PROTOCOL_ID: int
CONCENTRATION: real
DATE_PREPARED: datetime
PREPARED_BY: varchar(32)
REL_GENE_EXPR_RESULT_TYPE
ID: int
NAME: varchar(32)
ANALYSIS
ID: int
ALGORITHM_ID: int
PROTOCOL_ID: int
ANALYST_ID: varchar(32)
SCHEME_ID: int
ANALYSIS_DATE: datetime
DATA_SET_COLLECTION_ID: int
NAME: varchar(64)
ABS_GENE_EXPR_ATOM_RESULT
ANALYSIS_ID: int
UNIT_IDX: int
BLOCK_IDX: int
ATOM_IDX: int
BACKGROUND_INTENSITY: real
FLAG: int
PROTOCOL PARAMETERS
MEASUREMENT_ELEMENT_RESULT
ANALYSIS_ID: int
LOCATION_X: int
LOCATION_Y: int
INTENSITY: real
STATISTIC: real
PIXELS: int
INTENSITY_ORIG: real
FLAG: int
TEMPLATE_TYPE
ID: int
NAME: varchar(32)
PROTOCOL_TEMPLATE
ID: int
TYPE_ID: int
NAME: varchar(32)
PROTOCOL
ID: int
TEMPLATE_ID: int
PARAMETER_TYPE
ID: int
NAME: varchar(32)
PARAMETER_TEMPLATE
PROTOCOL_TEMPLATE_ID: int
PARAMETER_IDX: int
UNITS_ID: int
TYPE_ID: int
NAME: varchar(32)
STRING_VALUE: varchar(255)
PARAMETER
PROTOCOL_ID: int
PARAMETER_IDX: int
STRING_VALUE: varchar(255)
PARAMETER_UNITS
ID: int
NAME: varchar(32)
GATC Schema with Mapping and Spotted tables
Copyright 2001/2002
Affymetrix, Inc.
All Rights Reserved

FIGURE 5

FIGURE 6


600
Start
605
Direct excitation beam to a plurality of pixel locations on a probe array having a plurality of probe locations
610
Detect an emission signal having one or more emission values
615
Generate a plurality of pixel data based, at least in part, on the emission values
615
Generate a plurality of pixel data based, at least in part, on the emission values
620
Analyze the pixel data to generate intermediate results
625
Store the pixel data, the intermediate results, or both, in one or more data structures
Collect more data?
Yes
No
Process Data?
Yes
No
630
Provide one or more code libraries to enable transfer of user data from a user-provided software application directly or indirectly to the data structures
635
Compile executable code from the code libraries
640
Provide server executables to interface between the executable code and the data structure
645
Call the executable code from the user-provided software application
699
End